# *Japanese Character Printers*
# *EPL2 Programming*
# *Manual Addendum*

This addendum contains information unique to Zebra Technologies Japanese character bar code printers. The Japanese configuration printers support printer programming specified in the EPL2 (Page Mode) Programmer's Manual with additions of a modified **A** command, **b** Command (2D bar code), **i** Command (Inter-character Spacing) and the Japanese Character code pages. The commands are described in more detail on the following pages.

---

***Asian Character Sets Fonts 8 & 9***

The Page Mode EPL2 programming language supports up to two (2) font sets of a single Asian language as well as the standard EPL2 Latin (Multilingual) fonts 1-5 and downloadable soft fonts (A-Z and a-z). All fonts can be expanded both horizontally and vertically. The Asian fonts can also be printed from top to bottom or in the Latin character orientation from left to right. The Asian printers support the standard Latin fonts with the single character map code page 437 for all five (5) fonts.

The Asian characters are 16 bit (or double byte) mapped characters. The printed Asian character is dependent on the double byte ASCII values. The Latin (English, etc.) font sets are 8 bits per (or single byte) ASCII character maps.

---

***QR Code Bar Code***

The QR Code is only offered with Japanese Character bar code printer configurations.

---

**Only One (1) Asian language is supported by a printer.** Each Asian language (character set) is a separate printer firmware version.

Asian printers with flash firmware can be reprogrammed for a different Asian language, but we do not recommend this for normal use.

---

# A Command - ASCII Text

**Description** Prints a character string.

**Syntax** $Ap_1,p_2,p_3,p_4,p_5,p_6,p_7,$"DATA"

**Parameters**

$p_1$ = Horizontal start position (X) in dots.

$p_2$ = Vertical start position (Y) in dots.

$p_3$ = Rotation (Orientation)

| Value | Description |
|-------|-------------|
| 0 | No rotation,  Left to Right |
| 1 | 90 degrees,  Left to Right |
| 2 | 180 degrees,  Left to Right |
| 3 | 270 degrees,  Left to Right |
| 4* | 270 degrees,  Top to bottom |
| 5* | No rotation  Top to bottom |
| 6* | 90 degrees,  Top to bottom |
| 7* | 180 degree,  Top to bottom |
| **\* - Fonts 8 & 9 only** | |

$p_4$ = Font selection

| Value | Description | |
|-------|-------------|--|
| | **203 dpi** | **300 dpi** |
| 1 | 20.3 cpi, 6 pts, (8 x 12 dots) | 25 cpi, 4 pts, (12 x 20 dots) |
| 2 | 16.9 cpi, 7 pts, (10 x 16 dots) | 18.75 cpi, 6 pts, (16 x 28 dots) |
| 3 | 14.5 cpi, 10 pts, (12 x 20 dots) | 15 cpi, 8 pts, (20 x 36 dots) |
| 4 | 12.7 cpi, 12 pts, (14 x 24 dots) | 12.5 cpi, 10 pts, (24 x 44 dots) |
| 5 | 5.6 cpi, 24 pts, (32 x 48 dots) | 6.25 cpi, 21 pts, (48 x 80 dots) |
| 8 | 24 x 24 dots | Double-byte - 36 x 36 dots Single-byte - 24 x 36 dots |
| 9 | 16 x 16 dots | Double-byte - 24 x 24 dots Single-byte - 16 x 24 dots |
| A - Z | Reserved for Soft Fonts | |
| Note 1: *Fonts 1-5 are the standard EPL2 expandable fixed pitch (size) dot fonts.* Note 2: The Asian character sets use the **(i)** command to set the inter-character space between printed characters of an **(A)** command text data string. | | |

$p_5$ = Horizontal multiplier, expands printed text horizontally. Range: 1, 2, 3, 4, 5, 6, and 8.

$p_6$ = Vertical multiplier, expands printed text vertically. Range: 1 - 9.

$p_7$ = **N** for normal or **R** for reverse image

**"DATA"** = Represents a fixed data field.

The backslash (\) character designates the following character is a literal and will encode into the data field. Refer to the following examples:

| To Print | Enter into data field |
|----------|----------------------|
| " | \" |
| "Company" | \"Company\" |
| \ | \\ |
| \code\ | \\code\\ |

*Examples:*
```
↵
N↵
S1↵
D12↵
A50,0,0,1,1,1,N,"Example 1"↵
A50,50,0,2,1,1,N,"Example 2"↵
A50,100,0,3,1,1,N,"Example 3"↵
A50,150,0,4,1,1,N,"Example 4"↵
A50,200,0,3,2,2,R,"Example 5"↵
A50,280,0,8,2,2,N,"Example 6"↵
A50,350,0,8,1,1,N,"Example 7 ▓▒▓▒▓"↵
P1↵
```

*Will Produce:*

```
Example 1

Example 2

Example 3

Example 4

EXAMPLE  5

Example   6

Example   7   布步簿部
```

*Note: As shown in the example above, font 5 (example 5) only support the upper case characters.*

The data field can be replaced by or combined with the following commands/functions:

**Vnn** = Prints the contents of variable "**nn**" at this position where **nn** is a 2 digit number from 00 to 99.

**Cn** = Prints the contents of counter "**n**" at this position where n is a one digit number from 0 to 9.

**TT** = Prints the current time at this position in the predefined format. See the **(TT)** command for format selection and is only available if the printer has the RTC (Time & Date) option installed.

**TD** = Prints the current date at this position in the predefined format. See the **(TD)** command for format selection and is only available if the printer has the RTC (Time & Date) option installed.

*Examples:*

A50,0,0,1,1,1,N,"DATA"↵    : Prints quoted string
A50,50,0,2,1,1,N,V01↵    : Insert variable and print
A50,100,0,3,1,1,N,C1↵    : Insert counter and print
A50,150,0,4,1,1,N,TT↵    : Insert current time and print
A50,200,0,5,1,1,N,TD↵    : Insert current date and print

*or a combination of several options:*

A50,300,0,3,2,2,R,"Deluxe"V01C2"Combo"TDV01TT↵

: Writes the text "Deluxe" followed by the contents of variable 01
: followed by the contents of counter 2 followed by the text "Combo"
: followed by the current date followed by the contents of variable 01
: followed by the current time.

# A Command - ASCII Text

***Special JIS Code Page Programming Considerations*** The JIS keyboard technique utilizes double byte data characters. Each byte of the of the double byte character can be represented by one of the lower 128 ASCII data characters (20 through 7F hexadecimal).

The printer uses a Shift-& to recognize a JIS character text data string. Once the JIS text mode has been set with the Shift-&, then the JIS text mode remains in effect until the text data string is terminated.

**Shift** = 1C hexadecimal or 28 decimal

**&** (Ampersand) = 26 hexadecimal or 38 decimal

If any text data string contains the ASCII character values for the quote (") character, then it must be proceeded by a backslash.

**"** (quote) = 22 hexadecimal or 34 decimal

\ (backslash) = C5 hexadecimal or 92 decimal

***Tip:*** Enter the "Shift" character with the number pad on the standard 101 key PC keyboard. With the Number Lock on, type 028 on the number pad while holding the ALT key down. This is a standard method of entering ASCII characters not directly supported by a keyboard.

Note: When using the JIS, if the last character in a string of characters is a backslash (\), then the Shift character followed by any character is required to exit the JIS character mode.

***Example:*** A50,0,0,8,1,1,N,"Example JIS <Shift>&5\<Shift>."
: <Shift> represents the ASCII value
: 28 decimal.
: <Shift>& enters JIS text mode.
: <Shift>. exits JIS text mode
: (returns to Shift JIS text mode).
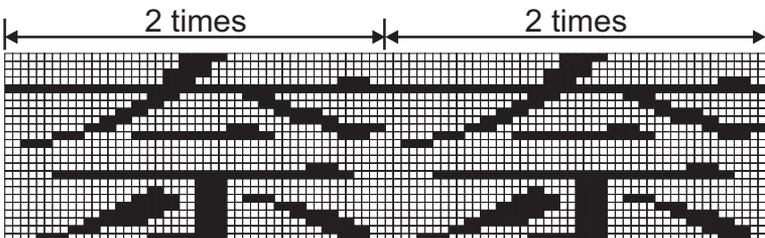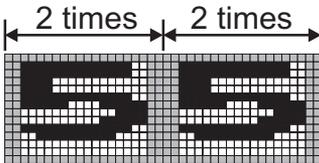
# A Command - ASCII Text

The Latin font (1-5) characters are dot mapped differently than the Asian font (8 & 9) characters. The Asian character does not have a built-in inter-character gap. The Latin characters include a single dot boarder around each character.

Note: 203 dpi characters shown as example
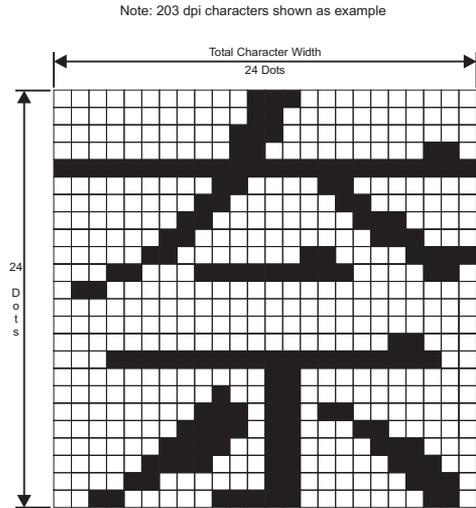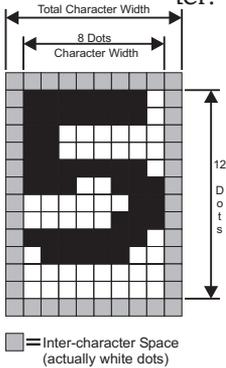


**Fonts 1 (8 x 12 dots)**          **Font 8 (24 x 24 dots)**

With the **(A)** command's horizontal multiplier (p5) set to 2, the inter-character spacing will look like the following example.

# A Command - ASCII Text

The Asian fonts can print character strings oriented from top to bottom (p*3* values 4-7), as well as the standard English word orientation from left to right (p3 values 0-3). The characters will print in the sequence that they are entered into the **(A)** command's data field.
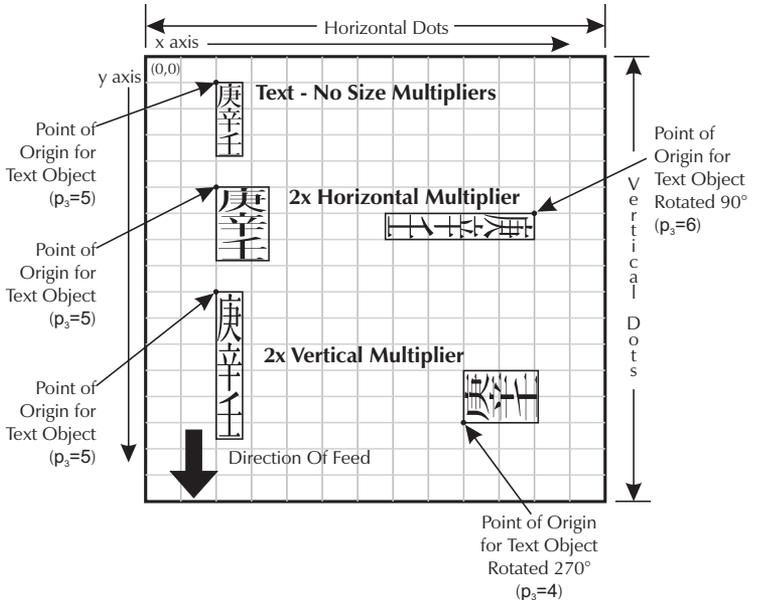


The printer automatically recognizes single byte characters and double byte characters in the Shift-JIS codepage. The single byte font characters are mapped to hexadecimal address range 00 to 7F hex (0-127 decimal) for 203 dpi printers and 00 to FF hex (0-255 decimal) for 300 dpi printers.

The double byte font characters are mapped to hexadecimal address ranges (see below).

| Codepage Reference | Range | From | | To | |
|---|---|---|---|---|---|
| | | 1st byte | 2nd Byte | 1st byte | 2nd Byte |
| JIS | 2120-7424 | 33 (21h) | 32 (20h) | 116 (74h) | 36 (24h) |
| Shift-JIS | 8140-9FFC | 129(81h) | 64 (40h) | 159 (9Fh) | 252 (FCh) |
| | E040-EAA4 | 224 (E0h) | 64 (40h) | 234 (EAh) | 164 (A4h) |

# i Command – Asian Character Spacing

*Description*   Places an adjustable inter-character space between Asian font characters, fonts 8 and 9, only. The inter-character spacing gets multiplied with the text string by the selected font's horizontal and vertical multiplier values.

*Note: At the time of print, the printers are not capable of using the* **i** *command within forms.*
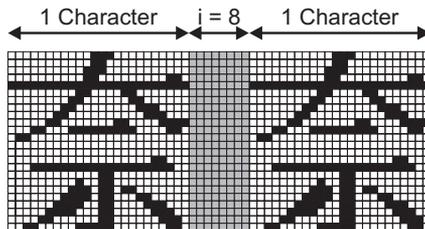
*Syntax*   **ip$_1$**

*Parameters*   **p$_1$** =   Space in dots between Asian characters.
Range: 0-9 (dots)
Default: 0 (dots or no space)

(**i**) Command
Parameter Set to Default (0 dots)

1 Character        1 Character
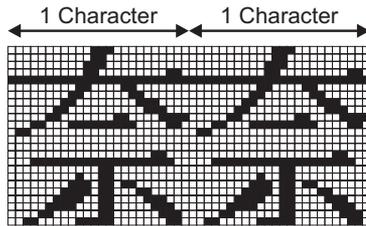


(**i**) Command
Parameter Set to 8 (8 dots)

1 Character        i = 8        1 Character

# **b** Command - 2D Bar Code - QR Code Specific Options

*Description* Use this command to generate QR Code bar code symbols with a single command. QR Code is only available in Japanese printer models. See the AIM web site for QR Code specifications at:
http://www.aimi.org/

*Syntax* $\mathbf{bp_1,p_2,p_3,[p_{4\text{-}9}]}$,**"DATA"**

*Parameters* $\mathbf{p_1}$ = Horizontal start position (X) in dots

$\mathbf{p_2}$ = Vertical start position (Y) in dots

$\mathbf{p_3}$ = **Q** - Must be "Q" for QR Code

---

Parameters $\mathbf{p_4}$ through $\mathbf{p_8}$ are optional and may be omitted. Default values will automatically be inserted when a parameter is omitted. Each parameter value must be proceeded by its associated command prefix character. The parameters $\mathbf{p_4}$ through $\mathbf{p_8}$ can be inserted into the command string in any order prior to the **"DATA"** and following $\mathbf{p_3}$. Commas between parameters $\mathbf{p_4}$ through $\mathbf{p_8}$ are not required.

$\mathbf{p_4}$ = Code Model (prefix **m**)

| Value | Description |
|-------|-------------|
| **1** | Model 1 |
| **2** | Model 2 - **Default** |

$\mathbf{p_5}$ = Scale Factor (prefix **s**)
**Default: 3**
**Range: 1-99**

$\mathbf{p_6}$ = Error Correction Level (prefix **e**)

| Value | Description |
|-------|-------------|
| **L** | Lower error correction, most data |
| **M** | **Default** |
| **Q** | Optimized for error correction over data |
| **H** | Highest error correction, least data |

$\mathbf{p_7}$ = Data Input Mode (prefix **i**)

| Value | Description |
|-------|-------------|
| **A** | Automatic Data Select - **Default** |
| **M** | Initializes the manual data mode and the data type is set by the first character in the fixed data field (**"DATA"**). |

$p_8$ = Append Symbol (prefix **D**)
The Append Symbol parameter option allows the programmer to join data from 2 to 16 QR code symbols.

| Sub-prefix | Values | Description |
|:---:|:---:|:---|
| **c** | 01-16 | Symbol Number |
| **d** | 01-16 | Divisions |
| **p** | 00-FF Hex. | Parity |

**"DATA"** = Represents a fixed data field.

Data sent to the printer is converted to one of four formats depending upon the value set by parameter $p_7$, Data Input Mode select. By default, the printer will automatically select the data mode for the entire fixed data string. The printer will check and change the data encoding method to achieve the highest data compression.

If parameter $p_7$ is set to **IM**, then first character in the **Data** must be one of the following:

**N** - **Numeric** (0-9)

**A** - **Alphanumeric** ( 0-9, A-Z, a-z  and space, $, %, *, +, -, . , /, : )

**K** - **Kanji** (Shift JIS character ranges 8140-9FFC and E040-EAA4 Hex)

**B** - **Binary**

The data field has reserved characters that normally can not be used within the data string, they are: **"** and **/**

The backslash (**\**) character designates the following character is a literal and will encode into the data field. Refer to the following examples:

| To Print | Enter into data field |
|:---|:---|
| " | \" |
| "Company" | \"Company\" |
| \ | \\ |
| \code\ | \\code\\ |

| | |
|---|---|
| *Fonts 1-5* | Fonts 1-5 are single byte, ASCII characters. See the EPL2 programmer's manual for fonts and codepages supported by the printer. |

| | |
|---|---|
| *Fonts 8 and 9*<br>*Single Byte*<br>*Character Map* | The single byte font characters are mapped to hexadecimal address range 00 to 7F hex (0-127 decimal). Font 8 are ASCII, code page 437, single byte characters. See below for the character map. |

```
   0  -      ☻ ☻ ♥ ♦ ♣ ♠ • ▪    ▫ ♪ ♀    ♫ ✿
  16  -
  32  -    !  "  #  ¥  %  &  '  (  )  *  +  ,  -  .  /
  48  -  0  1  2  3  4  5  6  7  8  9  :  ;  <  =  >  ?
  64  -  @  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O
  80  -  P  Q  R  S  T  U  V  W  X  Y  Z  [  \  ]  ^  _
  96  -  `  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o
 112  -  p  q  r  s  t  u  v  w  x  y  z  {  |  }
```

| | |
|---|---|
| *Fonts 8 and 9*<br>*Double Byte*<br>*Ideographic*<br>*Characters* | The Japanese ideographic font characters are mapped to double byte hexadecimal addresses. See the following pages for the codepage character mappings. |

203 dpi printers - JIS

203 dpi printers - Shift JIS

300 dpi printers - JIS

300 dpi printers - Shift JIS